# Smoothness, Synthesis, and Sampling: Re-thinking Unsupervised Multi-View Stereo with DIV Loss

Alex Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer

University of California, Santa Barbara, Santa Barbara CA, USA
{anrich, noahstier, psen, tholleren}@ucsb.edu

**Abstract.** Despite significant progress in unsupervised multi-view stereo (MVS), the core loss formulation has remained largely unchanged since its introduction. However, we identify fundamental limitations to this core loss and propose three major changes to improve the modeling of depth priors, occlusion, and view-dependent effects. First, we eliminate prominent stair-stepping and edge artifacts in predicted depth maps using a clamped depth-smoothness constraint. Second, we propose a learned view-synthesis approach to generate an image for photometric loss, avoiding the use of hand-coded heuristics for handling view-dependent effects. Finally, we sample additional views for supervision beyond those used as MVS input, challenging the network to predict depth that matches unseen views. Together, these contributions form an improved supervision strategy we call *DIV* loss. The key advantage of our DIV loss is that it can be easily dropped into existing unsupervised MVS training pipelines, resulting in significant improvements on competitive reconstruction benchmarks and drastically better qualitative performance around object boundaries for minimal training cost.

**Keywords:** Unsupervised Learning · Multi-View Stereo · Depth Prediction

## 1 Introduction

Multi-view stereo (MVS) is a fundamental problem in computer vision [10, 33, 35], with applications from augmented reality to autonomous driving and robot navigation. In recent years, MVS depth prediction using fully-supervised deep learning has seen great advances [1, 7, 13, 24, 28, 32, 47, 48]. While this has led to new breakthroughs on numerous benchmark datasets [4, 18, 22, 34, 50], these methods rely on accurate ground-truth 3D geometry collected with a depth sensor. This limits their training to mainly indoor settings on highly-constrained datasets.

A popular line of work aims to remove this restrictive 3D supervision requirement by training fully-unsupervised MVS networks [2, 5, 6, 16, 19, 23, 30, 40–42, 46, 54], taking an essential step toward scaling to large, diverse, and unlabeled image datasets. However, upon experimentation we have identified fundamental flaws with the core unsupervised loss function that has become the de facto standard in this field, which leads us to revise its basic assumptions. The result of our work is a novel loss formulation that is widely applicable as a drop-in replacement in unsupervised MVS training,

https://alexrich021.github.io/div-loss/

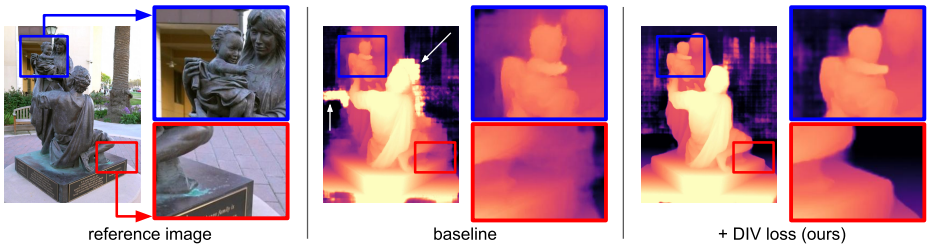


**Fig. 1:** Our DIV loss results in substantially more precise object boundaries and reduced artifacts when training unsupervised multi-view-stereo networks. We improve the handling of edges and view-dependent effects to produce more accurate 3D reconstructions with greatly improved visual quality. Results are from the **DIV-MVS** pipeline and corresponding baseline (see Sec. 4).

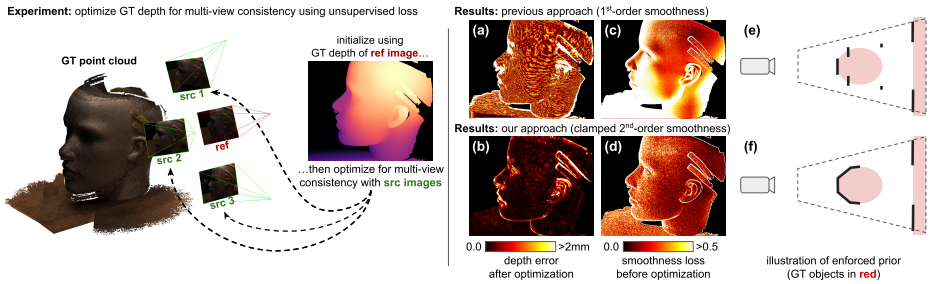


**Fig. 2: A motivating experiment.** Initializing from ground truth, we optimize a depth map for consistency with a set of images using the standard unsupervised MVS loss. If the loss is well-formulated, the depth map should remain unchanged, as it started from the optimal initialization. However, the result shows considerable deviation, with large errors at object boundaries and a pronounced stair-step effect (a). The artifacts stem from the depth-smoothness loss, which improperly penalizes the ground-truth depth, preferring locally-constant depth, and which is not sufficiently down-weighted to allow depth discontinuities at object boundaries (c). This results in a depth prior which does not properly model objects in the scene (e). Our improved smoothness loss properly models the scene (f), reducing the penalty on the ground truth depth (d) and largely eliminating the artifacts (b), showing it is a fundamentally more accurate objective.

resulting in drastically better qualitative performance around object boundaries and significant quantitative improvements on competitive reconstruction benchmarks for minimal training cost. See Fig. 1 for example qualitative improvements.

Our work was motivated by a simple experiment, shown in Fig. 2. Starting from ground-truth initialization, we optimize a single depth map for consistency with a set of images using the standard unsupervised loss. If the loss is well-formulated, the depth map should remain unchanged, since it started from the optimal initialization. However, after optimization we observe significant errors: depth bleeding across object boundaries and depth stepping instead of smooth surfaces (Fig. 2a). These artifacts are caused by a depth-smoothness loss which enforces a sub-optimal prior (Fig. 2e). The loss, a penalty on the $1^{st}$-order depth gradient which is down-weighted at object boundaries, has two issues. First, it encourages depth to be a series of fronto-parallel planes because they have a $1^{st}$-order gradient of 0, thereby locally minimizing the loss. Second, depth points bleed between objects to prevent the depth gradient from overpowering the down-weighting at object boundaries. We argue that a piecewise-planar prior is better

than a fronto-parallel one. One obvious method of encouraging this is to use a $2^{\text{nd}}$-order penalty on the depth gradient, as in some existing works [5, 16]. However, we find that the $2^{\text{nd}}$-order penalty incorrectly enforces smooth curvature across object boundaries instead of allowing sharp discontinuities, exacerbating bleeding between objects and harming performance. Our key insight is to *clamp* the $2^{\text{nd}}$-order depth gradient to some maximum value before applying the loss penalty. This truncates the magnitude of the penalty across boundaries, preventing large gradients from overpowering the down-weighting and therefore permitting sharp discontinuities where required (Fig. 2f). As in previous work, we use high image gradients to identify object boundaries, though we note this ignores the rare case where ground-truth depth discontinuities coincide with low image gradients. Our novel clamped depth smoothness greatly reduces artifacts in our experiment (Fig. 2b), confirming our intuition.

These surprising results indicate the standard loss has fundamental limitations, and motivate us to revisit its image-synthesis component as well. Previous methods handle view-dependent effects heuristically, by warping a set of supervision images to the reference image and computing a per-pixel loss against the reference image. For every pixel, loss is propagated only for the $K$ supervision pixels with the minimum loss among all warped supervision pixels. Given that 3D geometry is complex and difficult to model, we hypothesize a learning-based approach will outperform this heuristic approach. To this end, we propose to train a network that learns to identify and handle view-dependent effects. This network takes as input the set of warped supervision images, which can be thought of as multiple synthesized reference images, and outputs a weight map for each. A single synthesized reference image is then generated as a weighted sum of warped supervision images and used for image-synthesis loss computation, avoiding the use of hand-coded heuristics.

Finally, we observe that all previous approaches use the same set of images both as input to the MVS network *and* as supervision views in image-synthesis loss computation. We propose sampling additional views for supervision beyond those used as MVS input. We hypothesize that this additional challenge, forcing the network to predict depth matching unseen views, will lead to a representation that is more generalizable and robust to missing information. In summary:

1. We propose a novel depth-smoothness loss which properly enforces piecewise planarity of depth maps using penalization of the clamped $2^{\text{nd}}$-order gradient.
2. We propose a novel learning-based method for supervision by image synthesis, which greatly improves handling of view-dependent effects.
3. We propose to use additional views for supervision beyond those used as MVS input, challenging the network to predict depth that matches unseen views.

Together, these **D**epth-smoothness, **I**mage-synthesis, and **V**iew-sampling methods form an improved supervision strategy we call **DIV** loss. This loss can be used with existing unsupervised pipelines as a drop-in replacement for the previous loss functions. It is lightweight, requiring minimal additional GPU memory and runtime during training. MVS networks trained with our loss achieve state-of-the-art results among unsupervised methods on the DTU [18], Tanks and Temples [22], and ScanNet++ [51] datasets. Our predicted depth maps show clear and distinct object boundaries, leading to substantially sharper and visually cleaner 3D reconstructions.

## 2 Related Work

**Fully-Supervised MVS:** Many fully-supervised MVS depth-prediction methods have been proposed [9, 15, 17, 25, 32, 39, 48, 52]. While recurrent [49], optimization [53], and point-cloud methods [3] have been proposed for decreasing memory requirements, coarse-to-fine methods [13,28,47] are the most popular. Recent work focuses on extracting better 2D features [1, 7, 24]. Despite large improvements, supervised MVS methods are still constrained by their reliance on ground-truth 3D geometry.

**End-to-End Unsupervised MVS:** Khot *et al*. [19] first proposed a combination of depth-smoothness and min-$K$ image-synthesis losses for fully-unsupervised MVS. Nearly all unsupervised methods use a similar framework; they rely heavily on the core losses proposed by Khot *et al*., and focus instead on adding additional constraints to *complement* these core losses. MVS$^2$ [5] uses a cross-view consistency loss. M$^3$VSNet [16] adds a feature loss. JDACS [41], RC-MVSNet [2], and CL-MVSNet [40] all propose additional training branches with differing augmentation methods and consistency losses. Our work improves the core unsupervised loss which is critical for all of these unsupervised MVS methods, and thus would benefit all of them.

**Multi-Stage Self-Supervised MVS:** Some methods train an initial MVS network in an unsupervised fashion, then produce pseudo-depth for self-supervised training using this network. Geometric filtering and meshing [46], dense 2D optical flow correspondences [42], and filtering and probabilistic encoding [8] for pseudo-depth verification have been proposed. These frameworks rely on an initial unsupervised learning phase. Therefore our improvements are beneficial to all methods in this line of work as well.

**Depth Smoothness:** The depth-smoothness loss used in unsupervised MVS comes from unsupervised monocular depth estimation [11,12,27]. Most methods in both MVS and monocular depth estimation penalize the 1$^{\text{st}}$-order depth gradient [2, 11, 12, 14, 19, 21, 27, 37, 41]. Some methods penalize the 2$^{\text{nd}}$-order gradient [5, 16]; however, we find this actually harms performance. This is likely why 2$^{\text{nd}}$-order smoothness is not common in unsupervised MVS, despite being an intuitively better depth prior. Our proposed depth-smoothness loss instead penalizes the *clamped* 2$^{\text{nd}}$-order depth gradient. This clamping supports piecewise-planar depth with sharp object boundaries and, as a result, notably boosts performance. While we apply our loss to MVS, it is likely also beneficial for monocular depth estimation (though experimental evidence is needed).

## 3 Methods

In this section, we describe our novel DIV loss for unsupervised MVS (Fig. 3). Our formulation takes as input a reference image $\mathbf{I}$ with corresponding intrinsic and extrinsic camera parameters $\{\mathbf{K}, \mathbf{T}\}$, $N$ supervision images $\{\mathbf{I}_n\}_{n=1}^{N}$ each with camera parameters $\{\mathbf{K}_n, \mathbf{T}_n\}$, and a predicted depth $\mathbf{D}$ for the reference image $\mathbf{I}$ that is output by an MVS network to be trained. Other than differentiability, we make no assumptions on the manner in which $\mathbf{D}$ is predicted.

Our novel loss formulation is as follows. First, we apply a novel clamped 2$^{\text{nd}}$-order depth-smoothness loss to $\mathbf{D}$ (Sec. 3.2). The clamping mechanism is key to the performance boost from our depth smoothness. It allows for sharp discontinuities where

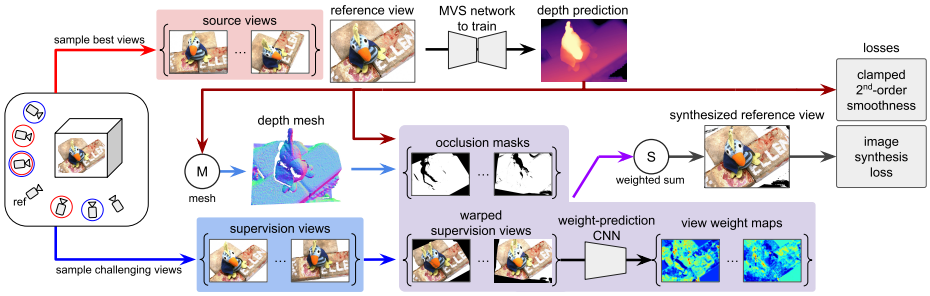


**Fig. 3: Overview of DIV loss for unsupervised MVS.** First, we apply a novel depth-smoothness loss to the predicted depth map. Second, we compute an image-synthesis loss against a synthesized reference image. The image is synthesized as an occlusion-aware weighted sum of warped supervision images, with weights predicted by a network and occlusion masks computed using existing methods [43]. Finally, when sampling supervision views, we include additional, challenging views beyond those used as MVS input.

required and, when used with the 2nd-order penalty, properly enforces piecewise-planar depth. Second, we apply a novel image-synthesis loss using a weighted combination of warped supervision images (Sec. 3.3). Supervision images are inverse-warped to the reference. The warped images, denoted $\{\hat{\mathbf{I}}_n\}_{n=1}^N$, are input to a network and per-pixel weight maps $\{\mathbf{W}_n\}_{n=1}^N$ are predicted. Following existing work [43], occlusion masks $\{\mathbf{M}_n\}_{n=1}^N$ are rendered via shadow-mapping [38]. A synthesized reference image, denoted $\hat{\mathbf{I}}$, is generated using weight, occlusion, and warp information, and standard image-synthesis losses are applied. We include additional supervision views beyond those used as MVS input via sampling of a larger image set (Sec. 3.4). We first describe the depth-smoothness and image-synthesis losses in previous work.

### 3.1 Unsupervised MVS Preliminaries

In the standard unsupervised MVS loss [19], the goal is to predict depth $\mathbf{D}$ that is locally smooth and maximizes image consistency when warping supervision views to the reference, as no ground-truth depth is available. The depth-smoothness loss used in the previous literature is:

$$L_{smooth} = \sum_{i \in [x,y]} \sum_{\mathbf{p}} e^{-\|\nabla_i \mathbf{I}(\mathbf{p})\|} |\nabla_i \mathbf{D}(\mathbf{p})| \tag{1}$$

This encourages depth discontinuities to coincide with large image gradients, as this implies the existence of an edge. Note it also encourages locally-constant depth.

To compute the image-synthesis losses, supervision images are inverse-warped to the reference. A reference pixel $\mathbf{p}$ is warped to $\hat{\mathbf{p}}$ in supervision view $n$ as follows:

$$\hat{\mathbf{p}} = \mathbf{K}_n \mathbf{T}_n \mathbf{T}^{-1} (\mathbf{D}(\mathbf{p}) \mathbf{K}^{-1} \mathbf{p}) \tag{2}$$

The warped supervision image $\hat{\mathbf{I}}_n$ is synthesized using bilinear sampling of $\mathbf{I}_n$ at the warped pixel location, i.e., $\hat{\mathbf{I}}_n(\mathbf{p}) = \mathbf{I}_n(\hat{\mathbf{p}})$. A photometric loss is computed as:

$$L_{photo} = \sum_{\mathbf{p}} \sum_{n \in \mathcal{K}(\mathbf{p})} \ell_{photo}(\hat{\mathbf{I}}_n(\mathbf{p}), \mathbf{I}(\mathbf{p})) \tag{3}$$

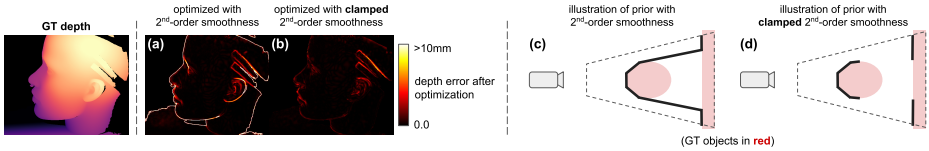


**Fig. 4: Another simple experiment:** We visualize the need for gradient clamping. As in Fig. 2, we optimize a GT depth map for consistency with a set of images. Now we use standard 2nd-order depth smoothness (Eq. 6) and our clamped 2nd-order smoothness (Eq. 7). Standard 2nd-order smoothness results in large edge errors (a, c) which harm network performance, while clamped 2nd-order smoothness permits sharp discontinuities where required (b, d) and boosts performance.

where $\ell_{photo}(\hat{\mathbf{c}}, \mathbf{c}) = \|\hat{\mathbf{c}} - \mathbf{c}\| + \|\nabla\hat{\mathbf{c}} - \nabla\mathbf{c}\|$. Crucially, $\mathcal{K}(\mathbf{p})$ indexes the minimum $K$ losses among warped views for a pixel $\mathbf{p}$. This discards high-loss outliers, presumed to be caused by occlusion or specularity. A structural similarity (SSIM) loss is computed using only the first 2 warped supervision images, $\hat{\mathbf{I}}_1$ and $\hat{\mathbf{I}}_2$:

$$L_{SSIM} = \sum_{\mathbf{p}} \sum_{n=1}^{2} \Big(1 - \text{SSIM}(\hat{\mathbf{I}}_n, \mathbf{I})(\mathbf{p})\Big) \tag{4}$$

The final loss is a weighted combination of these 3 terms:

$$L_{total} = \lambda_1 L_{photo} + \lambda_2 L_{SSIM} + \lambda_3 L_{smooth} \tag{5}$$

### 3.2 Improving the Depth-Smoothness Loss

As can be seen in Fig. 2, the smoothness loss in Eq. 1 encourages locally-constant depth with high error at object boundaries, resulting in artifacts. An immediate solution is to replace the 1st-order depth gradient in Eq. 1 with a 2nd-order gradient as in previous work [5,16], thereby enforcing locally-smooth depth rather than locally-constant depth:

$$L_{smooth} = \sum_{i,j \in [x,y]^2} \sum_{\mathbf{p}} e^{-\|\nabla_j \mathbf{I}(\mathbf{p})\|} |\nabla^2_{ij} \mathbf{D}(\mathbf{p})| \tag{6}$$

Note $i, j$ indexes all components of the 2nd-order gradient of $\mathbf{D}$. This loss reduces stair step artifacts, but exacerbates the bleeding between objects (see Fig. 4) and, as a result, harms performance. This bleed effect is consistent with enforcing smoothness across object boundaries. Solving it requires a mechanism that allows the smoothness constraint to be automatically relaxed across boundaries whose locations are not known a priori. To achieve this, we *clamp* the 2nd-order gradient to a maximum value of $\alpha$ prior to supervision, effectively truncating the magnitude of the gradient penalty in boundary regions:

$$L_{smooth} = \sum_{i,j \in [x,y]^2} \sum_{\mathbf{p}} e^{-\|\nabla_j \mathbf{I}(\mathbf{p})\|} \min(|\nabla^2_{ij} \mathbf{D}(\mathbf{p})|, \alpha) \tag{7}$$

Using the optimization experiment described in Sec. 1, we empirically set $\alpha = 4.0$.

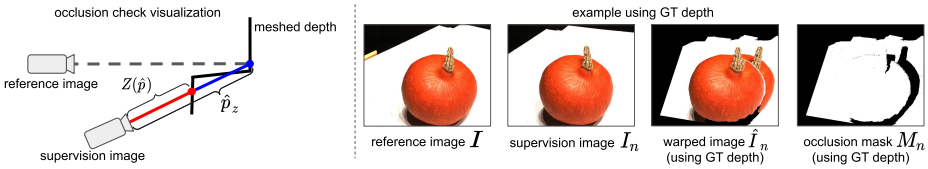


**Fig. 5: Occlusion-mask generation**. Following Xu *et al*. [43], we compute occlusion of the supervision image w.r.t. the reference to identify artifacts clearly visible in the naïve warp. Each depth pixel is projected to world space, then re-projected to a supervision image. The re-projected depth $\hat{\mathbf{p}}_z$ is tested against a z-buffer depth $\mathbf{Z}(\hat{\mathbf{p}})$ rendered using a meshed depth map to determine occlusion. This allows the network to ignore occluded regions, improving the training signal.

### 3.3 Improving the Image-Synthesis Loss

Instead of using individual warped supervision images $\{\hat{\mathbf{I}}_n\}_{n=1}^N$ to compute the image-synthesis loss, we synthesize a single image $\hat{\mathbf{I}}$ as a weighted combination of warped supervision images.

**Per-Pixel Weight Prediction:** A small CNN predicts per-pixel weight maps $\{\mathbf{W}_n\}_{n=1}^N$ for each supervision image. It takes as input the warped supervision images $\{\hat{\mathbf{I}}_n\}_{n=1}^N$, concatenated along the channel dimension to form an image volume of dimension $3N \times H \times W$. It outputs an $N \times H/4 \times W/4$ volume. The weight map $\mathbf{W}_n$ is taken as the $n^{th}$ slice of this volume and then upsampled via bilinear interpolation to $H \times W$.

**Occlusion Mask Generation:** Occlusions of the supervision image w.r.t. the reference image result in visible artifacts in the warped supervision image, and can therefore be handled by the learned weighting scheme described in the previous section. However, existing work solves for warping artifacts directly via shadow mapping [38] using the predicted depth. We find that this method used in combination with our weight network simplifies the network task, and leads to improved results.

Specifically, we render occlusion masks $\{\mathbf{M}_n\}_{n=1}^N$ as in Xu *et al*. [43] (see Fig. 5). We first back-project every depth pixel in $\mathbf{D}$ to form a point cloud in world coordinates. We then mesh this point cloud, forming 2 triangles for every $2 \times 2$ patch of back-projected pixels. For a supervision image $\mathbf{I}_n$, we render a $z$-buffer $\mathbf{Z}_n$ using the mesh and known camera parameters $\mathbf{T}_n, \mathbf{K}_n$. Finally, we form our occlusion mask $\mathbf{M}_n$:

$$\mathbf{M}_n(\mathbf{p}) = \begin{cases} 0 \text{ if } (\mathbf{Z}_n(\hat{\mathbf{p}}) - \hat{\mathbf{p}}_z) < \epsilon \\ 1 \text{ otherwise} \end{cases} \tag{8}$$

where $\hat{\mathbf{p}}$ is computed using Eq. 2, $\hat{\mathbf{p}}_z$ is the $z$ component of $\hat{\mathbf{p}}$, and $\epsilon$ is a small tolerance for floating point errors.

**Reference Synthesis and Loss:** A pixel $\mathbf{p}$ in our synthesized reference image $\hat{\mathbf{I}}$ is:

$$\hat{\mathbf{I}}(\mathbf{p}) = \sum_{n=1}^{N} \mathbf{W}_n^{(o)}(\mathbf{p}) \hat{\mathbf{I}}_n(\mathbf{p}) \tag{9}$$

where $\mathbf{W}_n^{(o)}$ is an occlusion-aware weight map combining the predicted weight maps with the occlusion masks:

$$\mathbf{W}_n^{(o)}(\mathbf{p}) = \frac{\mathbf{M}_n(\mathbf{p})\mathbf{W}_n(\mathbf{p})}{\sum_{n=1}^{N} \mathbf{M}_n(\mathbf{p})\mathbf{W}_n(\mathbf{p})} \tag{10}$$

i.e., weights are masked for pixels the occlusion mask identifies as occluded, then normalized to sum to 1 per-pixel. We additionally compute a final mask $\mathbf{M}$, marking pixels which are occluded w.r.t. every supervision image:

$$\mathbf{M}(\mathbf{p}) = \begin{cases} 0 \text{ if } \sum_{n=1}^{N} \mathbf{M}_n(\mathbf{p}) = 0 \\ 1 \text{ otherwise} \end{cases} \tag{11}$$

Our updated image-synthesis losses are as follows:

$$L_{photo} = K \sum_{\mathbf{p}} \mathbf{M}(\mathbf{p}) \ell_{photo}(\hat{\mathbf{I}}(\mathbf{p}), \mathbf{I}(\mathbf{p})) \tag{12}$$

$$L_{SSIM} = 2 \sum_{\mathbf{p}} \mathbf{M}(\mathbf{p}) \Big(1 - \text{SSIM}(\hat{\mathbf{I}}, \mathbf{I})(\mathbf{p})\Big) \tag{13}$$

Note that we multiply by $K$ and 2 respectively, so our final loss weights $\lambda_1, \lambda_2$ and $\lambda_3$ remain constant when switching Eqs. 12 and 13 with Eqs. 3 and 4. This is a critical detail that allows DIV loss to be used seamlessly with existing unsupervised MVS pipelines.

### 3.4 Supervision View Sampling

Previous unsupervised MVS pipelines select $N$ images from the scene with the best view score [48] against the reference image, and use these for both MVS input and network supervision. This guarantees that the network sees all images used for supervision during the forward pass. We find that including views beyond those used as MVS input challenges the network to match unseen views and boosts performance, leading to a representation that is more generalizable and robust to missing information. Specifically, we select our supervision images $\{\mathbf{I}_n\}_{n=1}^{N}$ by sampling from the $M$ images with the highest view score, with sample weighting according to the view score. This helps selects informative views while occasionally including a challenging view according to the view score metric. On DTU we set $M = 10$. We note the performance boost from this method comes for free, since simply changing the views used in the image-synthesis loss requires zero additional computation.

### 3.5 Our Full Loss Formulation

Our **D**epth-smoothness, **I**mage-synthesis, and **V**iew-sampling (**DIV**) loss is as in Eq. 5, i.e., $L_{total} = \lambda_1 L_{photo} + \lambda_2 L_{SSIM} + \lambda_3 L_{smooth}$. However, we substitute in our improved depth-smoothness loss given in Eq. 7 and our learned, occlusion-aware loss terms given by Eqs. 12 and 13. We sample $\{\mathbf{I}_n\}_{n=1}^{N}$ as described in Sec. 3.4. When used with existing pipelines, we mirror the hyperparameters of previous work [2, 5, 16, 19, 40, 41]: $\lambda_1 = 12$ (or 8 for CL-MVSNet [40]), $\lambda_2 = 6$, and $\lambda_3 = 0.18$.

When used with a multi-resolution backbone like CasMVSNet [13], which predicts depth at multiple scales, we predict $\{\mathbf{W}_n\}_{n=1}^{N}$ for only the highest-resolution scale, then downsample for lower-resolution scales. We then compute occlusion masks $\{\mathbf{M}_n\}_{n=1}^{N}$ and occlusion-aware weight maps $\{\mathbf{W}_n^{(o)}\}_{n=1}^{N}$ for each scale. We reason that the salience of each view is independent of scale but the depth map self-occlusions are dependent on the depth map itself, which is predicted per-scale.

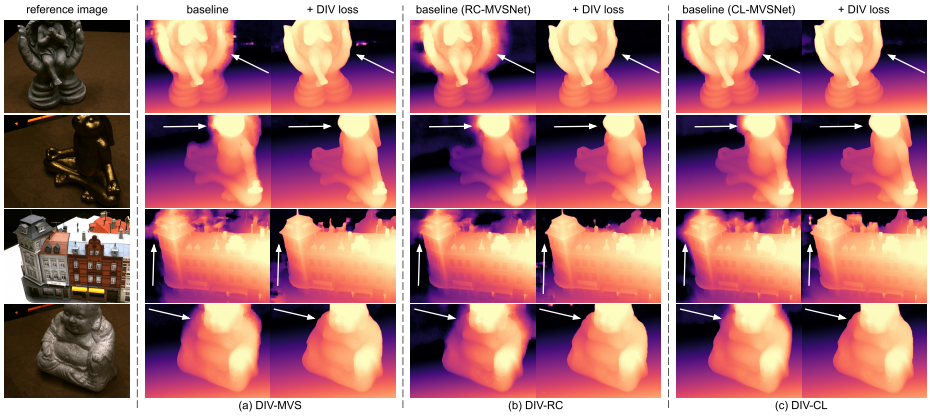


**Fig. 6: Qualitative depth results (DTU).** Representative results with and without our DIV loss for three different training pipelines. See Sec. 4 for a description of each pipeline. MVS networks trained with our loss produce depth maps with smooth and distinct foreground objects. Depth-gradient clamping permits sharp object boundaries, significantly reducing salient edge artifacts.

# 4 Experiments

## 4.1 Implementation Details

**Pipelines:** DIV loss can be used as a drop-in replacement in unsupervised MVS training pipelines of previous methods. To test our method, we conduct experiments on three pipelines we call **DIV-MVS**, **DIV-RC**, and **DIV-CL**, using a different depth-prediction network in each pipeline.

**DIV-MVS** is the main pipeline we conduct experiments on. In it, we use our loss formulation plus the data-augmentation loss proposed by Xu *et al*. [41]. We use Cas-MVSNet [13] with group-wise correlation [44] and the pixel-wise weight map for aggregating the cost volume as in Ding *et al*. [7]. **DIV-RC** is DIV loss with the exact RC-MVSNet [2] pipeline, i.e., we use the additional augmentation and neural radiance field training branches and CasMVSNet with variance aggregation. Likewise, **DIV-CL** is DIV loss with CL-MVSNet [40], i.e., we use the the additional contrastive-learning training branches and CasMVSNet with group-wise correlation.

All pipelines are implemented in PyTorch [29]. We use PyTorch3D [31] for the rendering step in occlusion-mask generation. We use Open3D [56] for visualization.

**Training Details:** We train using the DTU dataset [18], which consists of objects captured at 49 different camera positions under 7 lighting conditions. Following previous work [2,5,16,19,41], we use the pre-processed training set provided by Yao *et al*. [48]. We train all pipelines from scratch for 16 epochs using the Adam optimizer [20], with an initial learning rate of 0.0005. The learning rate is halved at epochs 10, 12, and 14. We use a batch size of 8. This requires 4, 8, and 8 NVIDIA RTX 3090 GPUs for **DIV-MVS**, **DIV-RC**, and **DIV-CL** respectively.

**Testing Details:** In addition to evaluation on the 22 scene DTU test set, we evaluate on the Tanks and Temples (T&T) intermediate and advanced test sets [22], and the ScanNet++ DSLR NVS validation set [51] without any fine-tuning. T&T consists of 14

| | Method | Acc. ↓ | Comp. ↓ | Ovr. ↓ |
|---|---|---|---|---|
| Supervised | CasMVSNet [13] | 0.325 | 0.385 | 0.355 |
| | CVP-MVSNet [47] | **0.296** | 0.406 | 0.351 |
| | AttMVS [26] | 0.383 | 0.329 | 0.356 |
| | PatchmatchNet [36] | 0.427 | 0.277 | 0.352 |
| | GeoMVSNet [55] | 0.331 | 0.259 | 0.295 |
| | MVSFormer-H [1] | 0.327 | **0.251** | **0.289** |
| Multi-Stage Self-Sup. | Self_sup CVP [46] | **0.308** | 0.418 | 0.363 |
| | U-MVS [42] | 0.354 | 0.354 | 0.354 |
| | KD-MVS [8] | 0.389 | **0.285** | **0.337** |
| E2E Unsup. | $M^3$VSNet [16] | 0.636 | 0.531 | 0.583 |
| | DS-MVSNet [23] | 0.374 | 0.347 | 0.361 |
| | JDACS-MS [41] | 0.398 | 0.318 | 0.358 |
| | ElasticMVS [54] | 0.374 | 0.325 | 0.349 |
| | RC-MVSNet [2] | 0.396 | 0.295 | 0.345 |
| | CL-MVSNet [40] | 0.375 | 0.283 | 0.329 |
| | **DIV-MVS (Ours)** | 0.382 | **0.279** | 0.330 |
| | **DIV-RC (Ours)** | 0.375 | 0.292 | 0.333 |
| | **DIV-CL (Ours)** | **0.362** | 0.280 | **0.321** |

**Table 1: DTU Dataset.** Point-cloud reconstruction metrics (in mm). Bold indicates best score in each section. **DIV-CL** outperforms all end-to-end unsupervised and multi-stage self-supervised methods on the Overall metric. Note that *lower* Acc. is better.

complex indoor and outdoor scenes while ScanNet++ consists of 50 complex indoor scenes. We use 5 input images of resolution $1600 \times 1184$ on DTU, 11 of resolution $1920 \times 1024$ on T&T, and 11 of resolution $1728 \times 736$ on ScanNet++ We use the photometric and geometric filtering point-cloud-fusion method used by Chang *et al*. [2] for DTU, T&T intermediate, and ScanNet++, and dynamic point-cloud fusion [45] for T&T advanced. On T&T, we find it beneficial to filter out depth pixels predicted with high confidence at the maximum depth plane, as these tend to correspond to objects beyond the depth plane.

### 4.2 Results

Overall, we find that DIV loss leads to much higher visual quality in both 2D depth maps and 3D reconstructions, as well as improved quantitative metrics, across all tested pipelines and datasets.

**DTU Results:** See Table 1 for point-cloud reconstruction metrics on the DTU test set, using the standard DTU metrics. Accuracy (Acc.) is the average distance in mm from each predicted point to its nearest ground-truth point. Completeness (Comp.) is the average distance in mm from each ground-truth point to its nearest predicted point. Overall (Ovr.), the average of accuracy and completeness, is the best measure of reconstruction quality. Note that counterintuitively, with the standard DTU metrics, a *lower* Acc. score is better.

**DIV-CL** outperforms all competing unsupervised and self-supervised methods and is competitive with many fully-supervised methods, achieving a 0.321 score (-0.008 vs. the best competing unsupervised method, CL-MVSNet). **DIV-MVS** and **DIV-RC** also achieve extremely competitive results, underscoring the effectiveness of DIV loss.

See Fig. 6 for qualitative depth-prediction results on DTU for all pipelines with and without DIV loss. Our loss greatly improves the visual quality of the depth in every case, producing sharp and accurate edges where previous work shows indistinct shapes with cloudy artifacts.

| | Method | DTU only | T&T intermed. F-score ↑ | T&T adv. F-score ↑ | ScanNet++ F-score ↑ |
|---|---|---|---|---|---|
| Supervised | CasMVSNet [13] | ✓ | 56.84 | 31.12 | - |
| | CVP-MVSNet [47] | ✓ | 54.03 | - | - |
| | AttMVS [26] | ✓ | 60.05 | 31.93 | - |
| | PatchmatchNet [36] | ✓ | 53.15 | 32.31 | - |
| | GeoMVSNet [55] | ✗ | 65.89 | 41.52 | - |
| | MVSFormer-H [1] | ✗ | **66.41** | **41.70** | - |
| Multi-Stage Self-Sup. | Self_sup CVP [46] | ✓ | 46.71 | - | - |
| | U-MVS [42] | ✓ | 57.15 | 30.97 | - |
| | KD-MVS [8] | ✗ | **64.14** | **37.96** | - |
| E2E Unsup. | $M^3$VSNet [16] | ✓ | 37.67 | - | - |
| | JDACS-MS [41] | ✓ | 45.48 | - | - |
| | DS-MVSNet [23] | ✓ | 54.76 | - | - |
| | ElasticMVS [54] | ✗ | 57.88 | 37.81 | - |
| | RC-MVSNet [2] | ✓ | 55.04 | 30.82 | 37.42 |
| | CL-MVSNet [40] | ✓ | 59.39 | 37.03 | 40.71 |
| | **DIV-MVS (Ours)** | ✓ | **60.36** | **38.35** | **41.64** |

**Table 2: T&T/ScanNet++.** Point-cloud evaluation results. Bold indicates best score in each section. "DTU only" indicates methods using no additional training data beyond DTU. **DIV-MVS** achieves SOTA results among unsupervised methods. Among methods which use only DTU for training, **DIV-MVS** even beats all multi-stage self-supervised and *fully-supervised* methods.

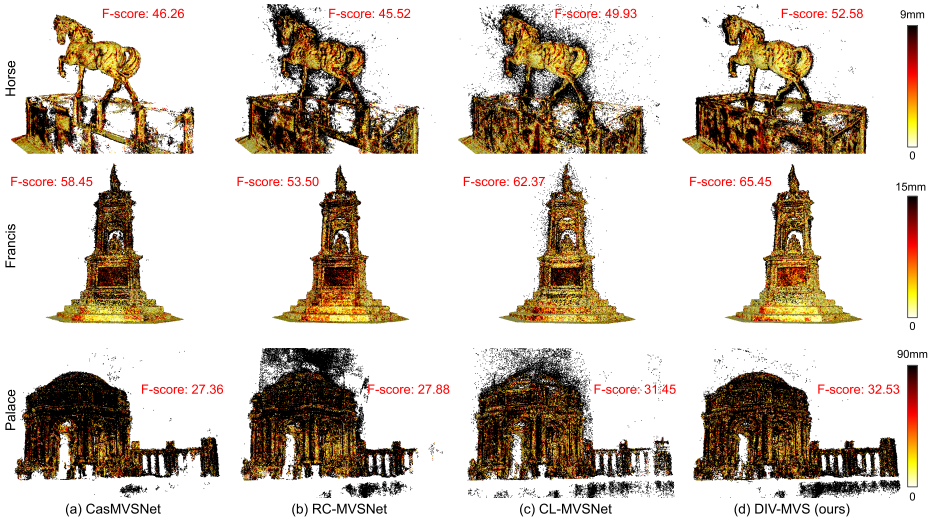


**Fig. 7: Qualitative reconstruction results (T&T outdoor scenes).** Darker regions indicate more error. **DIV-MVS** produces highly complete reconstructions with clean and accurate edges on these challenging, reflective and low-texture objects, achieving higher F-scores than both supervised (a) and unsupervised (b-c) baselines. This indicates improved generalization and robustness under difficult conditions. The large reduction in edge noise shows that improvements in object boundaries in depth predictions transfer to improvements in downstream reconstructions.

**T&T/ScanNet++ Results:** We apply **DIV-MVS** trained on DTU *with no fine-tuning* directly on the T&T and ScanNet++ datasets. See Table 2 and Figs. 1, 7, and 8 for quantitative and qualitative results. For T&T, we report mean F-score, provided by the online evaluation system and visualize all outdoor scenes. For ScanNet++, we report mean F-score using a 1cm threshold with an evaluation from Rich *et al*. [32].

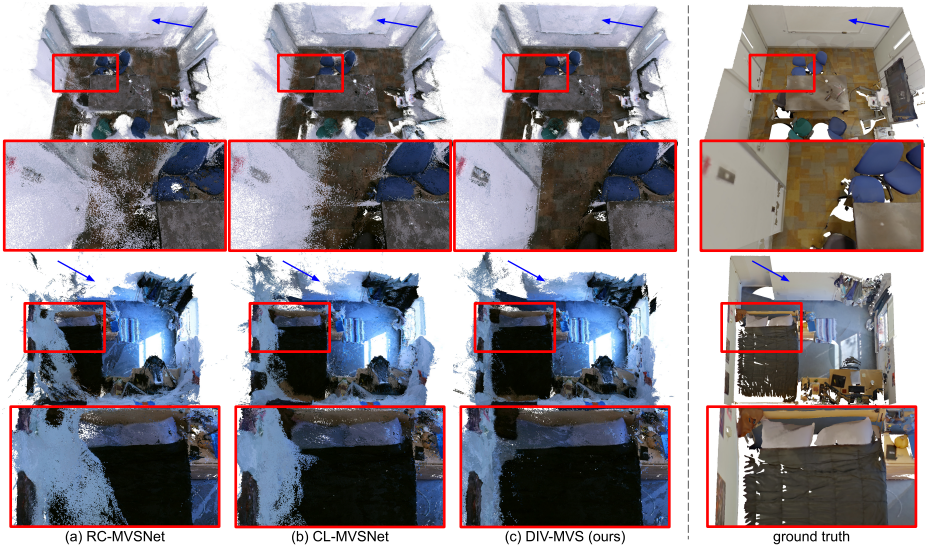


**Fig. 8: Qualitative reconstruction results (ScanNet++).** Comparisons between **DIV-MVS** (c) and the best competing unsupervised methods (a-b). As shown in the **red insets**, **DIV-MVS** produces *much* sharper object boundaries relative to unsupervised baselines, leading to a clean appearance and visually distinguishable objects. As identified by the **blue arrows**, **DIV-MVS** also yields better completion of textureless surfaces.

**DIV-MVS** achieves SOTA results on all datasets among unsupervised methods, outperforming even ElasticMVS [54], which uses additional training data. In fact, when considering only methods which use exclusively DTU for training, **DIV-MVS** even beats all multi-stage self-supervised and *fully-supervised* methods. These results show that models trained with DIV loss generalize effectively beyond the training distribution. On T&T, we find **DIV-MVS** has noticeably less per-point error and *drastically* reduced edge noise when compared to unsupervised baselines (see Fig. 7). This clearly shows the improvement in object boundaries in depth predictions transfers to improvements in downstream reconstructions. On ScanNet++, we find **DIV-MVS** reduces depth bleeding between blank background objects like walls and foreground objects like chairs and tables, relative to unsupervised baselines (see Fig. 8). We also observe that **DIV-MVS** completes textureless surfaces more reliably.

### 4.3 Ablation Study

**Our contributions are cumulative:** We analyze each component of DIV loss using **DIV-MVS** (Table 3). In addition to ablating our three main contributions (depth smoothness, image synthesis, and view sampling), we also test the occlusion masking with and without our learned image synthesis, and add special conditions for $1^{st}$-order smoothness with gradient clamping and $2^{nd}$-order smoothness without gradient clamping. See Fig. 9 for qualitative results. The baseline is defined in Sec. 3.1.

Our clamped $2^{nd}$-order smoothness improves the Overall score by -0.015 relative to the baseline. We also find that the combination of gradient clamping and $2^{nd}$-order

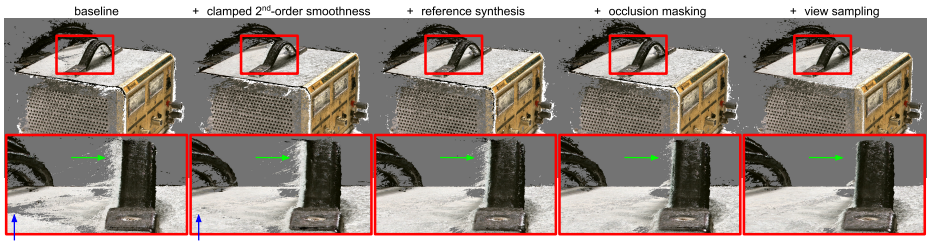


**Fig. 9: Visual ablation study (DTU).** We show ablation results using **DIV-MVS** (see Sec. 4.3 for details and Table 3 for metrics). **Dark gray** is background. As identified by the **green arrow**, our contributions noticeably reduce edge noise. This improvement is cumulative, together leading to much more accurate and visually clean results. As identified by the **blue arrow**, our clamped $2^{nd}$-order smoothness prior is largely responsible for filling holes in textureless regions, showing that it is a fundamentally better prior than $1^{st}$-order smoothness.

| depth smoothness | reference synthesis | occlusion masking | view sampling | Acc. ↓ | Comp. ↓ | Ovr. ↓ | Diff. |
|---|---|---|---|---|---|---|---|
| $1^{st}$-order | | | | 0.422 | 0.299 | 0.361 | +0.000 |
| clamped $1^{st}$-order | | | | 0.420 | 0.311 | 0.366 | +0.005 |
| $2^{nd}$-order | | | | 0.423 | 0.307 | 0.365 | +0.004 |
| clamped $2^{nd}$-order | | | | 0.398 | 0.294 | 0.346 | -0.015 |
| clamped $2^{nd}$-order | ✓ | | | 0.388 | 0.290 | 0.339 | -0.022 |
| clamped $2^{nd}$-order | | ✓ | | 0.401 | 0.302 | 0.351 | -0.010 |
| clamped $2^{nd}$-order | ✓ | ✓ | | 0.387 | 0.282 | 0.335 | -0.026 |
| clamped $2^{nd}$-order | ✓ | ✓ | ✓ | 0.382 | 0.279 | 0.330 | -0.031 |

**Table 3: DTU Dataset.** Ablation study for our **DIV-MVS** pipeline, showing that our contributions interact constructively. See Sec. 4.3 for details and Fig. 9 for qualitative results.

smoothness is critical. Clamping the $1^{st}$-order gradient results in +0.005 vs. baseline, as clamping likely exacerbates stair-stepping. $2^{nd}$-order smoothness without clamping results in +0.004 vs. baseline, as we observe this exacerbates bleeding between objects. This -0.019 difference in Overall score for $2^{nd}$-order smoothness with and without clamping is highly notable. The inclusion of learned reference synthesis improves the Overall score by an additional -0.007 without occlusion masks and -0.011 with. We also find occlusion masking without our learned reference synthesis actually harms metrics, indicating our synthesis method is critical. Finally, our supervision-view sampling improves the Overall score by an additional -0.005. In addition to improving the metrics, every component has a positive effect on 3D edge quality, and the clamped $2^{nd}$-order smoothness prior helps fill holes in textureless regions (see Fig. 9). These effects are also very prominent in the T&T and ScanNet++ qualitative results (Figs. 7 and 8).

**DIV loss is widely applicable:** In Table 4, we report reconstruction and depth metrics on DTU, and GPU memory during training for all pipelines with and without DIV loss. Abs. Depth Error is the mean absolute difference of ground-truth and predicted depth maps, in mm. For a fair comparison we re-train all methods from scratch and use identical point-cloud fusion parameters; the *only* difference is our loss formulation. DIV loss provides a performance boost across all pipelines on both the 3D reconstruction metric (Ovr.) and the depth metric (Abs. Depth Error) while requiring less than

| pipeline | DTU Ovr. ↓ without **DIV** | with **DIV** | diff | DTU Abs. Depth Error (mm) ↓ without **DIV** | with **DIV** | diff | Training Memory (GB) without **DIV** | with **DIV** | diff |
|---|---|---|---|---|---|---|---|---|---|
| **DIV-MVS** | 0.361 | 0.330 | -0.031 | 19.34 | 16.32 | -3.02 | 10.50 | 10.52 | +0.02 |
| **DIV-RC** | 0.350 | 0.333 | -0.017 | 21.76 | 21.01 | -0.75 | 12.24 | 12.26 | +0.02 |
| **DIV-CL** | 0.330 | 0.321 | -0.009 | 17.88 | 15.38 | -2.50 | 11.64 | 11.70 | +0.06 |

**Table 4: DTU Dataset.** Comparison of pipelines with and without our DIV loss. For fair comparison, we re-train all methods from scratch and use identical point-cloud fusion parameters. Abs. Depth Error is the mean absolute difference of ground-truth and predicted depth maps, in mm. Reported memory is for a batch size of 1. DIV loss boosts performance with negligible additional memory requirements for all pipelines.

0.1GB additional GPU memory during training. Rendering the occlusion masks only slightly increases the training runtime, from $1.81$ hrs. per epoch to $2.01$ hrs. per epoch for **DIV-CL**, which is a small cost of $\sim$3 hrs. relative to the $\sim$30 hrs. of total training time. Furthermore, **DIV-CL** achieves lower Abs. Depth Error on DTU after just a *single epoch* ($\sim$2 hrs. of training) than the *fully-trained* baseline ($\sim$30 hrs. of training). This shows that DIV loss is widely applicable as a drop-in replacement in unsupervised MVS training pipelines to increase reconstruction quality at minimal additional cost.

### 4.4 Limitations

**Depth smoothness:** In the relatively rare case where ground-truth depth discontinuities coincide with low image gradient, both DIV and previous smoothness losses will send a poor training signal, attempting to smooth the depth across the boundary. Further research is needed to flexibly encourage sharp depth edges in this case.
**Reference-view synthesis:** Our weight-prediction CNN cannot directly model specularity, as it has no camera direction information for any of the images. We find that it learns to simply, but effectively, down-weight specular reflections in supervision images. In the future, explicit modeling of view direction may yield improvements.

## 5 Conclusions

We have proposed a novel training strategy for unsupervised multi-view stereo called DIV loss, introducing three major innovations aimed at handling object boundaries and occlusion effects. First, our clamped $2^{\text{nd}}$-order smoothness constraint eliminates prominent stair-stepping and edge artifacts in predicted depth maps. Second, our reference-view synthesis learns from data to handle occlusion and view-dependent effects, rather than relying on the error-prone min-$K$ heuristic. Third, our view sampling selects additional views for supervision beyond those used as MVS input, challenging the network to predict depth that matches unseen views. Our formulation is widely applicable as a drop-in replacement in existing unsupervised MVS training pipelines, resulting in significant improvements on competitive reconstruction benchmarks, with drastically better qualitative performance around object boundaries for minimal training cost. Our insights on promoting smoothness while allowing sharp discontinuities may be applicable to other tasks with similar characteristics, such as single-view depth estimation and dense optical flow.

## Acknowledgments

Support for this work was provided by ONR grant N00014-23-1-2118 as well as NSF grants IIS-2211784 and IIS-1911230.

## References

1. Cao, C., Ren, X., Fu, Y.: MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth. Transactions on Machine Learning Research (2022)
2. Chang, D., Božič, A., Zhang, T., Yan, Q., Chen, Y., Süsstrunk, S., Nießner, M.: RC-MVSNet: Unsupervised multi-view stereo with neural rendering. In: European Conference on Computer Vision (2022)
3. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. In: International Conference on Computer Vision (2019)
4. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Conference on Computer Vision and Pattern Recognition (2017)
5. Dai, Y., Zhu, Z., Rao, Z., Li, B.: MVS2: Deep unsupervised multi-view stereo with multi-view symmetry. In: International Conference on 3D Vision (2019)
6. Darmon, F., Bascle, B., Devaux, J., Monasse, P., Aubry, M.: Deep multi-view stereo gone wild. In: International Conference on 3D Vision (2021)
7. Ding, Y., Yuan, W., Zhu, Q., Zhang, H., Liu, X., Wang, Y., Liu, X.: TransMVSNet: Global context-aware multi-view stereo network with transformers. In: Conference on Computer Vision and Pattern Recognition (2022)
8. Ding, Y., Zhu, Q., Liu, X., Yuan, W., Zhang, H., Zhang, C.: KD-MVS: Knowledge distillation based self-supervised learning for multi-view stereo. In: European Conference on Computer Vision (2022)
9. Düzçeker, A., Galliani, S., Vogel, C., Speciale, P., Dusmanu, M., Pollefeys, M.: DeepVideoMVS: Multi-view stereo on video with recurrent spatio-temporal fusion. In: Conference on Computer Vision and Pattern Recognition (2021)
10. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: International Conference on Computer Vision (2015)
11. Godard, C., Mac Aodha, O., Brostow, G.J.: Digging into self-supervised monocular depth prediction. In: Conference on Computer Vision and Pattern Recognition (2017)
12. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: International Conference on Computer Vision (2019)
13. Gu, X., Fan, Z., Dai, Z., Zhu, S., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Conference on Computer Vision and Pattern Recognition (2020)
14. Guizilini, V., Ambrus, R., Chen, D., Zakharov, S., Gaidon, A.: Multi-frame self-supervised depth with transformers. In: Conference on Computer Vision and Pattern Recognition (2022)
15. Hou, Y., Kannala, J., Solin, A.: Multi-view stereo by temporal nonparametric fusion. In: International Conference on Computer Vision (2019)
16. Huang, B., Yi, H., Huang, C., He, Y., Liu, J., Liu, X.: M3VSNet: Unsupervised multi-metric multi-view stereo network. In: International Conference on Image Processing (2021)
17. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: DPSNet: End-to-end deep plane sweep stereo. In: International Conference on Learning Representations (2019)
18. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: Conference on Computer Vision and Pattern Recognition. pp. 406–413. IEEE (2014)

19. Khot, T., Agrawal, S., Tulsiani, S., Mertz, C., Lucey, S., Hebert, M.: Learning unsupervised multi-view stereopsis via robust photometric consistency. arXiv preprint arXiv:1905.02706 (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
21. Klingner, M., Termöhlen, J.A., Mikolajczyk, J., Fingscheidt, T.: Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In: European Conference on Computer Vision (2020)
22. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)
23. Li, J., Lu, Z., Wang, Y., Wang, Y., Xiao, J.: DS-MVSNet: Unsupervised multi-view stereo via depth synthesis. In: ACM International Conference on Multimedia (2022)
24. Liao, J., Ding, Y., Shavit, Y., Huang, D., Ren, S., Guo, J., Feng, W., Zhang, K.: WT-MVSNet: Window-based transformers for multi-view stereo. In: Advances in Neural Information Processing Systems (2022)
25. Luo, K., Guan, T., Ju, L., Huang, H., Luo, Y.: P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo. In: International Conference on Computer Vision (2019)
26. Luo, K., Guan, T., Ju, L., Wang, Y., Chen, Z., Luo, Y.: Attention-aware multi-view stereo (2022)
27. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In: Conference on Computer Vision and Pattern Recognition (2018)
28. Mi, Z., Di, C., Xu, D.: Generalized binary search network for highly-efficient multi-view stereo. In: Conference on Computer Vision and Pattern Recognition (2022)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (2019)
30. Qiu, K., Lai, Y., Liu, S., Wang, R.: Self-supervised multi-view stereo via inter and intra network pseudo depth. In: International Conference on Multimedia (2022)
31. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
32. Rich, A., Stier, N., Sen, P., Höllerer, T.: 3DVNet: Multi-view depth prediction and volumetric refinement. In: International Conference on 3D Vision (2021)
33. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (2016)
34. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: International Conference on Intelligent Robot Systems (Oct 2012)
35. Tola, E., Strecha, C., Fua, P.: Efficient large scale multi-view stereo for ultra high resolution image sets. Machine Vision and Applications **23** (09 2011). https://doi.org/10.1007/s00138-011-0346-8
36. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: PatchmatchNet: Learned multi-view patchmatch stereo (2021)
37. Watson, J., Aodha, O.M., Prisacariu, V., Brostow, G., Firman, M.: The temporal opportunist: Self-supervised multi-frame monocular depth. In: Conference on Computer Vision and Pattern Recognition (2021)
38. Williams, L.: Casting curved shadows on curved surfaces. In: Conference on Computer Graphics and Interactive Techniques (1978)

39. Xi, J., Shi, Y., Wang, Y., Guo, Y., Xu, K.: RayMVSNet: Learning ray-based 1D implicit fields for accurate multi-view stereo (2022)
40. Xiong, K., Peng, R., Zhang, Z., Feng, T., Jiao, J., Gao, F., Wang, R.: CL-MVSNet: Unsupervised multi-view stereo with dual-level contrastive learning. In: International Conference on Computer Vision (2023)
41. Xu, H., Zhou, Z., Qiao, Y., Kang, W., Wu, Q.: Self-supervised multi-view stereo via effective co-segmentation and data-augmentation. In: AAAI Conference on Artificial Intelligence (2021)
42. Xu, H., Zhou, Z., Wang, Y., Kang, W., Sun, B., Li, H., Qiao, Y.: Digging into uncertainty in self-supervised multi-view stereo. In: International Conference on Computer Vision (2021)
43. Xu, L., Luo, Y., Luo, K., Wang, Y., Guan, T., Chen, Z., Liu, W.: Exploiting the structure information of suppositional mesh for unsupervised multiview stereo. IEEE MultiMedia **29**(1), 94–103 (2022). https://doi.org/10.1109/MMUL.2021.3139012
44. Xu, Q., Tao, W.: Learning inverse depth regression for multi-view stereo with correlation cost volume. In: AAAI Conference on Artificial Intelligence (2019)
45. Yan, J., Wei, Z., Yi, H., Ding, M., Zhang, R., Chen, Y., Wang, G., Tai, Y.W.: Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In: European Conference on Computer Vision (2020)
46. Yang, J., Alvarez, J.M., Liu, M.: Self-supervised learning of depth inference for multi-view stereo. In: Conference on Computer Vision and Pattern Recognition (2021)
47. Yang, J., Mao, W., Alvarez, J.M., Liu, M.: Cost volume pyramid based depth inference for multi-view stereo. In: Conference on Computer Vision and Pattern Recognition (2020)
48. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: MVSNet: Depth inference for unstructured multi-view stereo. In: European Conference on Computer Vision (2018)
49. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent MVSNet for high-resolution multi-view stereo depth inference. In: Conference on Computer Vision and Pattern Recognition (2019)
50. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. Conference on Computer Vision and Pattern Recognition (2020)
51. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: ScanNet++: A high-fidelity dataset of 3d indoor scenes. In: International Conference on Computer Vision (2023)
52. Yi, H., Wei, Z., Ding, M., Zhang, R., Chen, Y., Wang, G., Tai, Y.W.: Pyramid multi-view stereo net with self-adaptive view aggregation. In: European Conference on Computer Vision (2020)
53. Yu, Z., Gao, S.: Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In: Conference on Computer Vision and Pattern Recognition (2020)
54. Zhang, J., Tang, R., Cao, Z., Xiao, J., Huang, R., Fang, L.: ElasticMVS: Learning elastic part representation for self-supervised multi-view stereopsis. In: Advances in Neural Information Processing Systems (2022)
55. Zhang, Z., Peng, R., Hu, Y., Wang, R.: GeoMVSNet: Learning multi-view stereo with geometry perception. In: Conference on Computer Vision and Pattern Recognition (2023)
56. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)